

METHOD AND APPARATUS FOR ENHANCED FORWARD ERROR  
CORRECTION IN A NETWORK

5

**Cross Reference to Related Applications**

The present application is a continuation-in-part of co-pending U.S. application serial number 09/587,741, filed June 5, 2000, the teachings of which are incorporated herein by reference.

10

**Field of the Invention**

The invention relates to communications networks in general. More particularly, the invention relates to a method and apparatus for enhancing forward error correction (FEC) in a network such as a long-haul communications network.

15

**Background of the Invention**

The capacity of long-haul communication systems, such as "undersea" or "submarine" systems has been increasing at a substantial rate. For example, some long-haul optically amplified undersea communication systems are capable of transferring information at speeds of 10 gigabits per second (Gbps) or greater. Long-haul communication systems, however, are particularly susceptible to noise and pulse distortion given the relatively long distances over which the signals must travel (e.g., generally 600-10,000 kilometers). Forward Error Correction (FEC) is a technique used to compensate for this distortion and provide "margin improvements" to the system. The margin improvements can be used to increase amplifier spacing or increase system capacity. In a Wavelength Division Multiplexing (WDM) system, the margin improvement can be used to increase the bit rate of each WDM channel, or decrease the spacing between WDM channels thereby allowing more channels for a given

amplifier bandwidth. Accordingly, improvements in FEC techniques directly translate into increased capacity for long-haul communication systems.

- FEC coding can be described essentially as the incorporation of a suitable code into a data stream for the detection and correction of data errors about which there is no previously known information. Error correcting codes are generated for a stream of data (i.e., encoding) and are sent to a receiver. The receiver recovers the error correcting codes and uses them to correct any errors in the received stream of data (i.e., decoding).
- These deterministic codes can uniquely decode any errors in the data and consequently correct them, within certain constraints. The challenge is to find “suitable” codes that can be efficient in both complexity and cost for a given system.

There is a large number of error-correction codes, each with different properties that are related to how the codes are generated and consequently how they perform. Some examples of those codes are the linear and cyclic Hamming codes, the cyclic Bose-Chaudhuri-Hocquenghem (BCH) codes, the convolutional (Viterbi) codes, the cyclic Golay and Fire codes, and some newer codes such as the Turbo convolutional and product codes (TCC, TPC). The codes that are frequently used for application in high bit-rate communication systems, however, are a set of cyclic, non-binary, block codes known as Reed-Solomon (RS) codes.

Conventional long-haul communication systems typically use the “RS 255/239” error-correction code to perform FEC. The RS 255/239 error-correction code yields approximately 5 decibels (dB) coding gain with about 6.7% redundancy. Due to various engineering margins, beginning-of-life (BOL) Q of these FEC-enhanced systems is on the order of 15 dB.

This permits the design of systems with end-of-life (EOL) Q as small as 11.2 dB. The term "Q" refers to one measure of the signal-to-noise ratio (SNR) of a system.

Because nonlinear impairments are still the prevailing limitation of system capacity, a greater coding gain above that provided by RS 255/239 would allow for further capacity improvements. There are coding techniques that provide higher coding gains of 10 dB or higher. These coding techniques, however, need more than 100% signal redundancy and therefore higher line rates. Current long-haul communication systems are limited to line rates of approximately 12.5 Gbps, and therefore cannot take advantage of these coding techniques without sacrificing capacity. Furthermore, these coding techniques require a soft decision receiver that increases latency and costs for the system.

In view of the foregoing, it can be appreciated that a substantial need exists for an enhanced FEC method and apparatus that solves the above-discussed drawbacks and deficiencies.

### **Summary of the Invention**

One embodiment of the present invention comprises a method and apparatus to perform error correction. A stream of data is encoded using concatenated error correcting codes. The encoded data is communicated over a long-haul transmission system. The encoded data is decoded using the codes.

With these and other advantages and features of the invention that will become hereinafter apparent, the nature of the invention may be more clearly understood by reference to the following detailed description of the invention, the appended claims and drawings attached herein.

### Brief Description of the Drawings

FIG. 1 illustrates a system suitable for practicing one embodiment of the present invention.

FIG. 2 is a block diagram of a FEC encoder in accordance with one embodiment of the present invention.

FIG. 3 is a block diagram of a FEC decoder in accordance with one embodiment of the invention.

FIG. 4 is a block flow diagram of the operations performed by an FEC codec in accordance with one embodiment of the present invention.

FIG. 5 is a block flow diagram of an encoding process in accordance with one embodiment of the present invention.

FIG. 6 is a block flow diagram of a decoding process in accordance with one embodiment of the present invention.

FIG. 7 is an illustration of packing code blocks into a frame in accordance with one embodiment of the present invention.

FIG. 8 is an illustration of the interleaving process in accordance with one embodiment of the present invention.

FIG. 9 illustrates plots of the theoretical upper bounds showing BER versus Q in accordance with one embodiment of the present invention.

FIG. 10 illustrates a first set of plots of a theoretical error bound in accordance with one embodiment of the present invention.

FIG. 11 illustrates a second set of plots of a theoretical error bound in accordance with one embodiment of the present invention.

FIG. 12 illustrates a plot of simulation results against the theoretical error bound in accordance with one embodiment of the present invention.

FIG. 13 illustrates a plot comparing coding gains from various concatenated RS codes in accordance with one embodiment of the present invention.

FIG. 14 illustrates a plot of interleave depth versus coding gain in accordance with one embodiment of the present invention.

FIG. 15 is a block flow diagram of an encoding process in accordance with another embodiment of the present invention.

5 FIG. 16 is a block flow diagram of a decoding process in accordance with another embodiment of the present invention.

FIG. 17 is an illustration of packing code blocks into a frame in accordance with another embodiment of the present invention.

10 FIG. 18 illustrates plots of the theoretical upper bounds showing BER versus Q in accordance with another embodiment of the present invention.

### Detailed Description

The embodiments of the present invention include a method and 15 apparatus to increase coding gains in a long-haul communications system using concatenated error-correcting codes (“concatenated codes” or “product codes”). A long-haul communications system is defined herein to include any system designed to transport signals over a distance of greater than 600 kilometers. For example, a long-haul optically amplified 20 undersea communication system is typically engineered to carry signals from one continent to another (e.g., North America to Europe). Concatenated codes refer to the use of two or more levels of FEC coding. The performance improvement from concatenated codes arises from the fact that any residual errors from one level of decoding will be corrected in 25 the second level of decoding.

Concatenated codes are designed to have a strong first-level (inner) code (e.g.,  $t = 16$ ) and a weaker second-level (outer) code (e.g.,  $t = 8$ ), with an interleaving step inbetween the two. Interleaving re-distributes or

"spreads" the errors from an undecodable inner code block over several outer code blocks. The re-distribution or spreading of errors brings the average number of errors per code block to within the error-correction capability of the code at least at the outer decoding level. The interleaver  
5 provides an FEC coding improvement corresponding to the depth of interleaving ("interleave depth") as discussed below.

One embodiment of the present invention utilizes RS error correcting codes. An RS code word consists of a "block" of  $n$  "symbols",  $k$  of which represent the data, with the remaining  $(n - k)$  symbols  
10 representing the redundancy or check symbols. These check symbols are appended to the data symbols during the encoding step, and are used to uniquely detect and correct bit errors at the decoder, within the error-correction capability of the code. After the decoding operation, the check symbols are stripped from the block, and the corrected data symbols are  
15 obtained. The data symbols themselves are left unmodified during the encoding step, and it is for this reason that the RS code is referred to as a "systematic" code. The rate of the RS code is the ratio of data symbols (or equivalent, bits) to codeword symbols (or bits). The overhead of the code  
is the ratio of the check symbols to data symbols, i.e., the overhead -  
20  $((1/\text{rate}) - 1)$ .

The non-binary nature of block RS codes is manifest in the fact that a code symbol is not exactly a bit but rather it consists of several bits. The typical symbol size  $m$  is 8 bits, or a standard byte. The number of check symbols used determines the error-correction capability of a particular RS  
25 code. For example, a code that can correct  $t$  symbol errors in a block of  $n$  symbols requires at least  $2t$  check symbols, so that the number of data symbols that can be transmitted in this block is  $k = n - 2t$ . Furthermore, for a given symbol size  $m$ , the maximum number of symbols per block,  $n$ , has

to be less than or equal to  $2^m - 1$  to ensure unique decodability. For example, for  $m = 8$ , we have  $n = 255$ , and for  $t = 8$  symbol errors in this case, the maximum number of data symbols is  $k = 239$ . This is represented in compact form as a 255/239 ( $n/k$ ) RS code.

5        RS error correcting schemes also include the use of a shortened RS code. A shortened RS code is one where some of the data symbols are left unused. For example a shortened 223/207 RS code of length  $n^* = (n - s) =$  223 symbols transmits 207 data symbols in a block with error correction capability of up to 8 symbol errors. The disadvantage of shortened codes,  
10      relative to full-length codes, is that they are rate-inefficient. Some practical considerations, such as the maximum number of code-word symbols having to be  $n^*$  ( $< n$ ) in some cases, however, may actually require this form. Shortened codes are implemented in both software and hardware by transforming a  $(n-s)/(k-s)$  RS code to an  $n/k$  code by padding  $s$  dummy  
15      symbols (e.g., 0) before encoding. At the decoder, this operation is reversed. After decoding, the padded symbols are stripped from the block.

A desirable property of RS codes is that they are maximum-distance codes. This means there is sufficient uniqueness between codewords such that the maximum number of errors in the (encoded) message can be  
20      corrected, for a given amount of redundancy, without the occurrence of a decoding error. This directly reflects the efficiency of these codes.

The decodability of the RS code can be demonstrated with a brief example. If the bit-error rates (BER) of the transmission channel is such that only a single symbol error is expected ( $t=1$ ),  $2t$  check symbols are  
25      required. In the case of an 8-bit symbol ( $m=8$ ), this translates to 16 check bits. Of the 16 bits in this code, 8 bits are used to uniquely locate the symbol error (one out of  $2^8 = 256$  possibilities, corresponding to one out of 255 symbol positions, in addition to the error-free case). The remaining 8

bits are used to uniquely determine the error pattern (one out of  $2^8 = 256$  error patterns, including the error-free pattern). Various procedures for encoding and decoding RS codewords are well-known in the art, and therefore will not be further described herein.

5       The use of concatenated codes provides relatively powerful error correction with relatively little additional processing power. The overhead of a 2-level concatenated RS code can be calculated as  $(r_1.r_2)^{-1} - 1$ , wherein  $r_1$  and  $r_2$  are the rates of the inner and outer codes, respectively. The concatenated RS code itself can be represented in compact form as  $n_2/k_2 - n_1/k_1$ , where the subscripts 1 and 2 represent the inner and outer codes, respectively. Conventional FEC coding schemes (e.g., RS 255/239) provide a transmission performance improvement equivalent to Q-factor of about 5 dB while providing 7% extra bits as redundancy. One embodiment of the present invention uses a concatenated RS code that provides an additional 10 coding gain of approximately 2 dB while providing an extra 16% redundancy bits (a total of 23%). The embodiment uses an FEC encoder/decoder using a concatenated RS coding scheme with interleaving between the stages. More particularly, the FEC encoder/decoder utilizes a concatenated RS code of 223/207-255/223.

20      Because line rates are currently technically limited to 12.5 Gbps, concatenated rate-efficient block codes were examined assuming a linear-channel model with additive white Gaussian noise (AWGN). Under this assumption, concatenated block codes are available that perform at rates from 0.80 to 0.8333 (i.e., signal redundancy between 25 and 20%) and net 25 coding gains somewhere between 1.5 dB to 2.5 dB greater than the gain achieved with the conventional RS 255/239 code.

At least two important discoveries by the inventors were significant in implementing concatenated codes in long-haul communication systems.

The first was the recognition that concatenated codes having an inner code that is stronger (i.e., lower code rate) than the outer code (i.e., higher code rate) is particularly useful in such systems. The second was the recognition that the class of codes utilized for the concatenated code significantly impacted system design.

With respect to the second discovery, two types of combinations were considered particularly advantageous for the long-haul communication systems. The first combination comprised a bit-based BCH inner code and a byte-based BCH outer code (referred to herein as “BCH-RS concatenated code”). This is because bit-based BCH codes are good for more uniformly distributed errors while RS codes are good for “bursty” channels. When an inner decoder cannot correct all the errors on the line, it starts generating bursts that can then be effectively handled by the outer RS decoder. The second combination comprised a pair of RS codes (referred to herein as “RS-RS concatenated code”). RS codes having a range from  $t = 2$  to  $t = 16$  were examined, with  $t$  representing a code strength that is defined as the maximal possible number of corrected symbols per codeword. The examination revealed that the concatenation of two RS codes of different strength would be particularly effective for undersea systems, provided that the outer code is interleaved before it is concatenated with the inner code. Interleaving is a technique that is normally used to spread bursty errors among several consecutive codewords. In this case an interleaver is inserted between the two concatenated codecs so the inner and the outer decoding processes are statistically de-correlated. In general practice, the greater the interleave depth the better coding performance is gained.

The BCH-RS concatenated code and the RS-RS concatenated code each offers advantages according to the needs and constraints of a

particular system. For example, the BCH-RS concatenation is good for channels that are both uniform and bursty in nature. The RS-RS concatenation is particularly good for bursty environments. Consequently, the RS-RS concatenation is well-suited for undersea communications  
5 systems because undersea channels are more bursty in nature.

Another important aspect of implementing an enhanced FEC system concerns digital frame alignment and synchronization in a very noisy environment. This is an important implementation issue because the enhanced FEC must operate at BER values as high as  $5 \times 10^{-2}$ . The framing  
10 and synchronization strategies used in conventional FEC systems are inadequate for conditions where BER is greater than  $10^{-4}$ .

It is worthy to note that any reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is  
15 included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

Referring now in detail to the drawings wherein like parts are designated by like reference numerals throughout, there is illustrated in  
20 FIG. 1 a system suitable for practicing one embodiment of the invention. FIG. 1 is a block diagram of a long-haul communications network 100 comprising a communications transmitter/receiver ("transceiver") 102 and a transceiver 108 connected via a network 106. Transceivers 102 and 108 each include a FEC encoder/decoder ("FEC codec") 104 and a FEC codec  
25 110, respectively. In this embodiment of the invention, long-haul communications network 100 is a conventional long-haul optically amplified undersea communication system with the optical transceivers modified to operate with a novel FEC codec performing in accordance with

T344747 20200405502

a novel concatenated FEC coding scheme. Network 100 in general, and network 106 in particular, are designed to transport optical signals over distances greater than 600 kilometers.

FIG. 2 is a block diagram of a FEC encoder in accordance with one embodiment of the invention. FIG. 2 illustrates a FEC encoder 200 representative of the structure performing the concatenated encoding function of FEC codecs 104 and/or 110. FEC encoder 200 comprises a first encoder 204, an interleaver 206 and a second encoder 208. First encoder 204 is also referred to herein as an "outer encoder." Second encoder 208 is also referred to herein as an "inner encoder." The operation of FEC encoder 200 will be discussed in more detail below with reference to FIGS. 4-6 and accompanying examples.

FIG. 3 is a block diagram of a FEC decoder in accordance with one embodiment of the invention. FIG. 3 illustrates a FEC decoder 300 representative of the structure performing the concatenated decoding function of FEC codecs 104 and/or 110. FEC decoder 300 comprises a first decoder 304, a deinterleaver 306 and a second decoder 308. First decoder 304 is also referred to herein as an "inner decoder." Second decoder 308 is also referred to herein as an "outer decoder." The operation of FEC decoder 300 will also be discussed in more detail below with reference to FIGS. 4-6 and accompanying examples.

For purposes of clarity, the encoding structure and functionality (i.e., FEC encoder 200) is discussed separately from the decoding structure and functionality (i.e., FEC decoder 300). It can be appreciated, however, that both the encoding and decoding structure and functionality can be combined into a single FEC codec (e.g., FEC codecs 104 and 110) and still fall within the scope of the invention.

The operation of systems 100, 200 and 300 will be described in more detail with reference to FIGS. 4-6. Although FIGS. 4-6 presented herein include a particular sequence of steps, it can be appreciated that the sequence of steps merely provides an example of how the general functionality described herein can be implemented. Further, each sequence of steps does not have to be executed in the order presented unless otherwise indicated.

FIG. 4 is a block flow diagram of an FEC process 400 consistent with one embodiment of the invention. In this embodiment of the invention, 10 FEC encoder 202 performs the FEC encoding. A stream of data is encoded using concatenated error correcting codes at step 402. The encoded data is communicated over a long-haul transmissions system at step 404. In one embodiment of the invention the long-haul transmission system communicates the encoded data at least 600 kilometers. The encoded data 15 is decoded using the error correcting codes at step 406.

FIG. 5 is a block flow diagram of an encoding process in accordance with one embodiment of the invention. FIG. 5 illustrates an encoding process 500 that is representative of step 402 described with reference to FIG. 4. The stream of data is packed into a first frame of first blocks at step 20 502. The first frame is also referred to herein as an "unencoded outer frame." A first error correcting code is generated for each of the first blocks at step 504. The first error correcting codes are appended to the first blocks to create a second frame of second blocks at step 506. The second frame is also referred to herein as an "encoded outer frame." The second 25 frame of second blocks is packed into a third frame of third blocks at step 508. The third frame is also referred to herein as an "unencoded inner frame." A second error correcting code is generated for each of the third blocks at step 510. The second error correcting codes are appended to the

third blocks to create a fourth frame of fourth blocks at step 512. The fourth frame is also referred to herein as an “encoded inner frame.”

The first frame, second frame, third frame and fourth frame each have a predetermined length. In one embodiment of the invention, the 5 length of the second frame matches the length of the third frame. In this manner, no padding is required for the third frame. This decreases the latency associated with such padding hardware and techniques. In alternative embodiments, however, the length of the second frame is less than the length of the third frame. In such a case, the third frame is padded 10 with padding symbols until the length of the third frame matches the length of the second frame. In this case, the increase in FEC coding efficiency is sufficient to compensate for the latency incurred by padding.

The embodiments of the invention use interleaving during the encoding and decoding process. More particularly, the interleaving 15 operation occurs during the packing of the second blocks from the second frame into the third blocks of the third frame, and vice-versa. It can be appreciated, however, that the interleaving process can occur as a separate step from the packing process and still fall within the scope of the invention. The interleaving operation can be either bit interleaving or byte 20 interleaving. In one embodiment of the invention, the third frame has a 1- $N$  of third blocks, with  $N$  matching an interleave depth for the encoding process. In one advantageous embodiment  $N = 64$ , while in another  $N = 16$ .

The error correcting codes can be any code from a group comprising 25 the linear and cyclic Hamming codes, the cyclic BCH codes, the convolutional Viterbi codes, the cyclin Golay and Fire codes, and some newer codes such as TCC and TPC. The concatenated error correcting code pair may be separately represented as a first and second error correcting

code, with the first error correcting code represented as  $x/y$  and the second error correcting code represented as  $z/x$ . In one embodiment of the invention, the first correcting code is a reed-solomon (RS) code. More particularly, the first error correcting code is an  $x/207$  RS error correcting code. The second error correcting code is also a RS code. The second error correcting code is a  $255/x$  RS error correcting code. In one advantageous embodiment of the invention the  $x$  is equal to 223 symbols. This two level FEC coding results in a net coding gain of approximately 1.8 decibels while performing at a bit error rate of  $10^{-10}$ . This embodiment adds a redundancy percentage to the communicated encoded data of approximately 23 percent.

In an alternative embodiment of the invention, the first error correcting code is one of a group comprising a bit based BCH code and a byte based BCH code. The second error correcting is also one of a group comprising a bit based BCH code and a byte based BCH code. Further, the first error correcting code is stronger than the second error correcting code.

FIG. 6 is a block flow diagram of a decoding process in accordance with one embodiment of the invention. FIG. 6 illustrates a decoding process 600. The second error correcting codes and third blocks are recovered from the fourth blocks at step 602. The second error correcting codes are used to correct errors for the third blocks at step 604. The second blocks are unpacked from the third blocks at step 606. The unpacking process also includes a deinterleaving operation described below. The first error correcting codes and the first blocks are recovered from the second blocks at step 608. The first error correcting codes are used to correct errors for the first blocks at step 610.

The operation of systems 100, 200 and 300, and the flow diagram shown in FIGS. 4-6, can be better understood by way of example. A

software-based Monte-Carlo simulation was developed in the C programming language for fast processing of the encoding and decoding operations. As described above, the concatenated RS codes involve two independent levels of RS encoding (and decoding) with an interleaving (de-interleaving) step in between them.

FIG. 7 is an illustration of how code blocks are packed into a frame in the encoding step. An integral number of first blocks 702 at the first (outer) encoding level are packed into a first frame 704 (i.e., the unencoded outer frame). Check symbols 706 for first blocks 702 are generated by a first encoder (e.g., first encoder 204) of a FEC encoder (e.g., FEC codec 104 or FEC encoder 200). Check symbols 706 are appended to first blocks 702 to form second blocks 708. Second blocks 708 are packed into a second frame 710 (i.e., the encoded outer frame). The bits (or bytes) from second blocks 708 are interleaved, and they are packed into third blocks 714 of a third frame 712 (i.e., unencoded inner frame). In this example, second frame 710 and third frame 712 have the same length in terms of bits (or bytes), although the block size will likely vary between the two frames. In other words, third frame 712 is required to be an integral number of third blocks 714, the size of which is different from that of second blocks 708. Thus, in order for second frame 710 and third frame 712 to be of the same length, the number of second blocks 708 and third blocks 714 per frame in each of these frames, respectively, has to be chosen appropriately.

If second frame 710 and third frame 712 cannot be made to match with an integral number of blocks, third frame 712 is padded or “stuffed” with dummy symbols until they are of equal length. The padding process, however, represents an increase in latency in a hardware implementation, or increased processing time in software. In one embodiment of the invention, the lengths of the frames are therefore chosen to minimize the

number (or reduce to zero) of stuffed symbols, while at the same time keeping the number of second blocks per second frame to a minimum.

Once second blocks 708 from second frame 710 are packed and interleaved into third blocks 714 of third frame 712, check symbols 716 are generated for third blocks 714 by a second encoder (e.g., second encoder 208) of an FEC encoder (e.g., FEC codec 104 or FEC encoder 200). Check symbols 716 are appended to third blocks 714 to form a set of fourth blocks 718 of a fourth frame 720 (i.e., the encoded inner frame). Once the two-level encoding process is performed, the encoded data stream is communicated to a transceiver (e.g., transceiver 108) for decoding by a FEC decoder (e.g., FEC codec 110 or FEC decoder 300).

FIG. 8 is an illustration of the interleaving process in accordance with one embodiment of the invention. As shown in FIG. 8, interleaving between the two encoding steps discussed with reference to FIG. 7 (between packing the second and third frames) amounts to re-distributing the errors in bit-groupings or bytes that are either 1-bit or 8-bits long. FIG. 8 illustrates an example of byte interleaving after second frame 710 is encoded. The improvement in error correction is directly related to the depth of interleaving. Using the example illustrated in FIG. 8, full byte (or symbol) interleaving requires that each of the 223 symbols in each second block 708 (i.e., the outer frame) is re-distributed into 223 different third blocks 713 (i.e., the inner frame). In the case of full interleaving, the 223 symbols would require an interleave depth of 223 levels or 223 third blocks 714. If full bit interleaving were required in this case, each of the  $223 \times 8$  bits in each of second blocks 708 would be re-distributed into  $223 \times 8 = 1784$  different third blocks 714. In this case, the interleave depth is 1784 levels. Although full bit or byte interleaving improves the error correction, the

disadvantage of full interleaving is the large amount of memory required and the additional latency in a practical implementation.

Prior to evaluating the results of the software-based Monte-Carlo simulation described above, theoretical BER error bounds were established 5 to provide a basis for comparison. The theoretical BER error bounds estimate the maximum BER that is observed after error correction, using a particular code, of a message transmitted through a channel with a specific line BER. This served as a benchmark to ensure that both software-based and hardware-based codes were performing "correctly." The benchmark 10 also served as a way to efficiently evaluate and compare the performance of several different codes. The theoretical error bound was established using the following assumptions:

- 15 1. A Binomial distribution of un-correlated bit errors is observed on the channel (note that for  $\text{BER} < 10^{-1}$ , the Binomial distribution can be approximated by a Poisson distribution, and for a large number of events, or transmitted bits, the Binomial probability distribution can be approximated by a normal distribution under certain conditions that are valid in this case);  
20 2. No additional errors are committed at the decoder if it is found that a block is undecodable because the errors are passed through unchanged.  
25 3. Errors are equally likely to occur in the data and check symbols so that number of residual errors is reduced further when check symbols are stripped from the block; and  
4. For  $\text{BER} < 5 \times 10^{-2}$ , at most 2 bit errors per symbol error are likely to occur.

FIG. 9 illustrates plots of the theoretical upper bounds showing BER versus  $Q$  in accordance with one embodiment of the invention. The comparison with conventional error bounds indicates the estimate of the BER after error correction (1) follows the line or channel BER very closely when the error-correction capability is exceeded when the line BER is  $\sim 10^{-2}$ , and (2) is a less-conservative estimate of the maximum estimated BER after correction. The “looser” upper bound was subsequently justified by strong agreement with the results from the software-based Monte-Carlo simulation of the BER.

FIG. 10 illustrates a first set of plots of a theoretical error bound in accordance with one embodiment of the invention. The theoretical model was verified by evaluating the theoretical performance of single-level RS codes. For example, the use of 7% RS (FEC) codes yield a coding gain in  $Q$  of greater than 5 dB at an output BER level of  $10^{-10}$  over unencoded transmission. The legend in FIG. 10 also reflects a  $\delta Q$  reduction in the coding gain due to transmission at higher bit rates. The distinction between gross and net coding gain in  $Q$  is discussed with reference to FIG.

11

FIG. 11 illustrates a second set of plots of a theoretical error bound in accordance with one embodiment of the invention. The  $Q$  of the system is defined as usual, and any increase in  $Q$  due to error-correction coding is defined as coding gain. There is a difference, however, between a “gross gain” and a “net gain” in  $Q$ . More particularly, the gross gain does not account for the system impairment from the increased noise bandwidth, and consequent reduction in  $Q$ , due to transmission at higher line rates. The transmission performance plots shown in FIG. 11 thus indicate the gross coding gain, through a direct conversion from the BER after error correction to the system  $Q$  in dB. The loss in  $Q$ , however, is reflected

separately in the plots (in the legend), where  $\delta Q$  represents an adjustment to the coding gain as a function of the modified (higher) line rate due to the overhead – this then gives the net coding gain. This is shown to provide an estimate of the net gain that will be computed in an actual wet-system  
5 simulation that accounts for various system impairments such as nonlinearity in the fiber, inter-symbol interference, chromatic dispersion, and so forth.

FIG. 12 illustrates a plot of the simulation results against the theoretical error bound in accordance with one embodiment of the invention. As shown in FIG. 12, results of the simulation, for BER after error correction, compared extremely favorably to the theoretical error bounds. The agreement was good for single-level RS codes concatenated RS codes, and concatenated shortened RS codes. Furthermore, two separate C programs were independently developed, one for the Monte-  
10 Carlo simulation and other for incorporation into a system experiment, where the encoding, decoding, interleaving, de-interleaving, frame and PRBS-pattern synchronization, were software-based. The two programs yielded almost identical results for the BER improvement after error correction, confirming not only the correct implementation of the various  
15 algorithms, but also the robustness of the frame synchronization.  
20

In one embodiment of the Monte-Carlo simulation, the speed of the C-code is about 464 blocks per second of decoding on a 350 Megahertz (MHz), Pentium-II processor with 64 Megabytes (MB) of Random Access Memory (RAM). A single, random, encoded frame is re-sent several times  
25 through the channel until an encoded bit-stream of sufficient length is transmitted. The random noise introduced to each frame, however, is different as the computer's system clock is used to generate a seed for the (C) random-number generator (e.g., "*srand*").

FIG. 13 illustrates a plot comparing coding gains from the various concatenated RS codes in accordance with one embodiment of the invention. Several RS codes were evaluated with symbol size of 8 bits because this is the most common hardware implementation for RS coding.

- 5 This translates to a full code-block length of 255 symbols, each 8 bits long. Furthermore, overhead was constrained to a maximum of 23% since next generation terminal designs are limited to line rates of approximately 12.3 - 12.5 Gbps. Consequently, a set of concatenated RS codes having a net overhead (inner and outer codes) fixed at 23% (i.e.,  $t_1 + t_2 = t_0$ , where  $t_1$  and
- 10  $t_2$  are the number of symbol errors that can be corrected in the inner and outer codes, and their sum, which is the net error correction capability) was examined.

- 15 FIG. 14 illustrates a plot of interleave depth versus coding gain in accordance with one embodiment of the invention. The effect of the depth of primarily bit and (8-bit) byte interleaving provides varying results in terms of coding gain for the system. As shown in FIG. 13, deeper levels of byte interleaving (e.g., up to 223 levels for a 223/207-255/203 code) improve coding performance but with marginal gains beyond about 64 levels of byte interleaving. This is significant because deeper levels of
- 20 interleaving utilize more memory and increase latency without providing much coding gain. The loss in coding from byte-interleaving from 64 to 16 levels is about 0.27 dB at output BER levels of  $10^{-9}$ .

- 25 Byte interleaving yields better error-correction than bit-interleaving. This is because 1-2 bit-errors per symbol error at line BER of  $10^{-2}$  and smaller on average is demonstrated. This means that symbol (or byte) interleaving is desirable to spread out the residual bit errors from the first level of decoding (inner decoding). Bit-interleaving on the other hand may not re-distribute the residual bit errors "maximally" unless full bit

interleaving is implemented ( $223 \times 8$  levels). The disadvantage in this implementation then is that additional software processing time or hardware latency is required.

The net coding gain for the 223/207-255/223 concatenated RS code was evaluated. The reason this particular combination was singled out is that its inner and outer code block lengths are such that they can be efficiently packed into a frame that can be as short as one unencoded inner block, with  $k_1 = 223$  (or equivalently, one encoded outer block, with  $n_2 = 223$ ). Furthermore, this mode corrects up to 16 symbol errors in the inner code, and up to 8 symbol errors in the outer one - this is compatible with existing code encoder/decoder designs from LSI Logic which supports coding engines that can correct anywhere from 3 to 16 symbol errors in a block that is up to 255 symbol long. The net coding gain for this code at output BER of  $10^{-10}$  is estimated to be 1.8 dB, relative to the 7%- overhead RS 255/239 FEC code as illustrated in Table 1.

**Table 1**

Code Type	Overhead	Net Q (dB) Output BER = $10^{-10}$	Gross Q (dB) (Output BER - $10^{-10}$ )
No FEC	0%	16.08	16.08
RS 255/239 FEC	6.7%	11.03	10.75
223/207 - 255/223 EFEC*	23.2%	9.26	8.35
215/207 - 255/215 EFEC	23.2%	9.06	8.15

\* Enhanced FEC

RS concatenated code 215/207 - 255/215 was also considered because it provides better error-correction with the same overhead. This represents a stronger inner code which corrects up to 20 symbol errors, and a weaker outer code which corrects up to 4 symbol errors. Modifying the  
5 hardware design for  $t = 20$  by LSI Logic and ASIC International is possible to implement this type of code. The net coding gain for this code at output BER of  $10^{-10}$  is about 2 dB, relative to the 7%-overhead RS 255/239 FEC code (see Table 1). Other RS codes with symbol sizes other than 8 bits (e.g.,  
10 9 and 7 bit-long symbols) are possible, but may require modifications to existing core designs for the encoder and decoder.

The RS codes considered herein are rate-efficient codes with good error-correction performance in high bit-rate communication systems. The complexity of the encoding and decoding operations is also not too high so that a hardware implementation is both feasible and cost-effective. The  
15 constraint on the maximum overhead allowed for error correction (about 23%) is imposed by terminal hardware speeds. This limits systems, in one embodiment of the invention, to concatenated RS codes of the form  $x/207 - 255/x$ , where  $x$  is a measure of the asymmetry of the strengths of the inner and outer codes. One embodiment of the invention includes  $x$  to be 223  
20 because this is a good performing code based on existing core hardware designs for the encoder and decoder. The net coding gain for this code at an output BER of  $10^{-10}$  is estimated to be about 1.8 dB, relative to the 7%-overhead RS 255/239 FEC code.

Byte interleaving appears to be marginally better than bit-interleaving in terms of performance, but may have a greater impact on hardware designs in terms of reduced latency and a smaller memory requirement. The depth of byte interleaving can be limited to about 64 levels, and even to as few as 16 levels, without sacrificing much coding

gain. They may have significant impact on the architecture of the hardware in that up to 16 parallel coding engines can be accommodated on a single chip currently.

- The core designs for the coding engine can be modified to support
- 5 RS codes that can correct up to 20 symbol errors per block, or RS codes with symbol sizes of 9 and 7 bits. Consequently, other promising codes that offer additional coding gain are available. Other potential code types include a 3-level concatenated RS code, and a 2-level concatenated RS code that are further concatenated with bit-based codes such as BCH codes.
  - 10 These offer further improvement in error correction, with the disadvantage being “diminishing returns” on additional levels of coding, and increased latency. Finally, a class of codes known as Turbo codes provides superior performance as well.

Factors that may affect implementation of enhanced FEC coding as described herein include the effects of chromatic dispersion, Kerr non-linearity, and polarization fading. These effects will cause the noise properties of a communication channel to be different from the computer simulations and theory used above. The assumption used above is that the noise is AWGN causing binomially-distributed errors.

- 20 A testing platform can test EFEC over a real, long distance, optical channel. The test platform will test various EFEC codes by encoding and decoding in software. The optical channel is implemented by looping a short span of amplifiers (200-500 km) using standard techniques. The encoded data is generated by a computer program and loaded into a Bit Error Rate Test Set (BERTS) (8 Mb). After transmission through the loop, every sixteenth bit of the noisy data is acquired by a high-speed data acquisition unit. This data is stored on a hard disk or removable disk for subsequent data processing.
- 25

Well-generalized computer programs can generate properly encoded and framed data, as well as decoding the acquired data. These programs are capable of:

- Different Code Types: concatenated RS, BCH, and Turbo codes;
  - RS Codes with variable block length ( $n$ ), overhead ( $n - k$ ) and symbol size ( $m$ );
  - Frame alignment; user-specified Frame Alignment Word (FAW);
  - PRBS generation and re-synchronization with variable word length;
  - Interleaving; variable bit groupings (e.g., bit and byte interleaving) and variable number of blocks per frame;
  - Burst boundary detection and re-synchronization;
  - Error detection: i.e., the software acts as the receiving BERTS;
  - Roll control: independently determine the roll state for each burst of data from the loop;
  - Interleaving of 16 valid data streams for the transmitter; and
  - Interface to the BERTS.
- An example of FEC 104, FEC 110, FEC encoder 202 and FEC decoder 302, includes a modified RS code engine made by LSI Logic. These engines were developed for "100 K" LSI 0.8 micrometer Complementary Metal-Oxide Semiconductor (CMOS) process. The  $t = 8$  engine is also applied in conventional FEC Application Specific Integrated Circuits (ASICs). LSI Logic has since introduced three newer CMOS processes G10, G11 and G12. G12 process is the newest 0.18 micrometer geometry standard cell array with supply voltage options of 1.5 V, 2.5 V and 3.3 V depending on

processing speed. This particular process allows for integration on several million logic cells with high speed processing cores. It is anticipated that the G-12 serial processing speed could be greater than 1 Gbps and the interfaces could be made as fast as 2.5 Gbps. This would be sufficient to  
5 meet the requirements of at least one embodiment of the invention, which are in/out (I/O) interface speed of less than 780 Mbps and processing speed less than 390 Mbps. Because a large part of the processor logic circuit will operate at much lower speed, a majority of the cells could be powered with the low-voltage (1.8 V) option. Consequently, the power  
10 dissipation could be significantly reduced in comparison with the latest 2.5 Gbps FEC design on the G10 process.

Accordingly, at least one embodiment of the invention can be implemented utilizing the enhanced FEC 12.5 Gbps codec unit with two to four G12 ASICs that will include the framing logic, buffers, most of the  
15 timing functions, the overhead multiplex and processing. This would also include a rather deep interleaver, which is needed for de-correlation of the two concatenated coding processes. The compilers exist between 100 kilobytes (K) and G12 process which are suitable for one embodiment of the invention. Consequently, core engine implementation can be  
20 implemented through modification of the compiler. Further, the production cost could be markedly reduced and the reliability considerably improved through the capability of an ultra large scale of integration.

Other potential FEC codecs with similar or better performance than LSI Logic G-12 process include those designed by Motorola, Texas  
25 Instruments, IBM and so forth. Although these technologies could not use the LSI Logic core designs, other RS and BCH core designs are available and are just as robust. In particular, AHA Inc. makes a wide spectrum of

RS core designs with an option of erasure that would make a concatenated code more efficient.

In addition, it can be appreciated that RS and BCH core designs are available for implementation on programmable logic arrays (PLA). These 5 are referred to as the "Hammer Codes." PLA implementation of additional coding of the overhead bytes (bits), external to the high-speed payload codec, would be very useful.

The feasibility of frame alignment and framing at very high BER rate (BER > 10<sup>-2</sup>) for 10 Gbps payloads has also been evaluated. The evaluation 10 reveals that a number of different frames are possible. It appears, however, that shorter frames are more robust than longer frames. Further, it would be beneficial if the FAW and the associated overhead bits (OW and dedicated data channels) are not coded with the payload. Rather, they should be coded separately at much lower speed and possibly at a lower 15 code rate because there would be plenty of redundant bits available in a practical frame.

Optimal frame alignment methods and de-synchronization strategies have been explored. The studies indicate that the optimal FAW length is 16 bits, and that RS decoder engine "error diagnostics" should be 20 used to start a frame re-alignment process.

The testing of hardware core designs would also be beneficial in addition to the software approach of undersea channel coding tests described above. Low speed integrated circuits (ICs) for LSI Logic RS and BCS core designs are available. In addition, a low speed IC for an AHA RS 25 codec is also available. Potential test candidates include technology developed by Lockheed Martin Satellite division (previously Mount Whitney), which owns BCH and RS core designs on Vitesse Gallium

Arsenide (GaAs) gate arrays. The processing speed for these codecs is approximately 620 Mbps.

All these "codec ICs" are programmable, and need external frame alignment, First-In First-Out (FIFOs), and timing circuits. This calls for a 5 set of test boards. They should be developed to test ASIC prototypes to avoid problems in later implementation. The hardware testing boards are useful for system tests, both before and during the ASIC design phase. An initial design of low and high-speed test boards for LSI logic ICs that will be able to test concatenated RS schemes has already been established.

10        Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention. For example, although the embodiments 15 of the invention discuss a particular concatenated RS codec at the given signal redundancy constraint (<24%), it can be appreciated that additional coding gains may be achieved by concatenation of a RS and a punctured convolutional code, such as a TCC or TPC, or concatenation of BCH and RS codes. The problem with punctured convolutional schemes is that a soft- 20 decision receiver is required. The particular system design must take into consideration the difficulty of its implementation versus potentially superior performance (about 0.5 dB). Similarly, BCH core designs are not readily available and therefore may require additional implementation time.

25        Concatenation of a RS code and a turbo-code can yield as much as 10 dB coding gain, relative to the unencoded data, at only 26% signal redundancy. Turbo-codes are generally implemented using an encoder including at least two component codes separated by an interleaver. The

interleaver causes the encoders to be excited by two separate input sequences. Decoding of the turbo-code generally requires at least two separate decoders producing “soft” output. Decoding may be conducted iteratively, using information derived from the output of the first decoder  
5 to perform decoding in the second decoder, and vice-versa.

In an embodiment consistent with the invention including concatenation of a RS code and a turbo-code, the outer encoder 204 illustrated in FIG. 2 would be an RS encoder, and the inner encoder 208 would be a turbo-code encoder. Also, the inner decoder 304 illustrated in  
10 FIG. 3 would be a turbo-code decoder, and the outer decoder would be an RS decoder. The RS and turbo-codes used in such an embodiment may be of a variety of types. For example, the RS code may be a single or multi-level code and/or the turbo-code may be a TCC or a TPC. A TPC code is generally composed of a multi-dimensional array of block codes, such as  
15 Hamming and BCH codes. In the simplest configuration the constituent codes can consist solely of parity codes.

The operations performed by a communication system consistent with the invention for an embodiment including concatenated RS and turbo codes are the same as those illustrated generally in FIG. 4.

20 FIG. 15 is a block flow diagram illustrating an embodiment of an encoding process 1500 including concatenated RS and turbo codes that is representative of step 402 described with reference to FIG. 4. The stream of data is packed into a first frame of first blocks at step 1502. The first frame is also referred to herein as an “unencoded outer frame.” An RS error  
25 correcting code is generated for each of the first blocks at step 1504. The RS error correcting codes are appended to the first blocks to create a second frame of second blocks at step 1506. The second frame is also referred to herein as an “encoded outer frame.”

The second frame of second blocks is packed into a third frame of third blocks at step 1508. The third frame is also referred to herein as an “unencoded inner frame.” A turbo-code error correcting code is generated for each of the third blocks at step 1510 to create a fourth frame. The fourth 5 frame is also referred to herein as an “encoded inner frame.”

The first frame, second frame, third frame may each have a predetermined length, while the fourth frame may be viewed as having a predetermined dimensions depending on the type of turbo-code used. As described above with respect to FIG. 5, in one embodiment of the 10 invention, the length of the second frame may match the length of the third frame. In alternative embodiments, however, the third frame may be padded with padding symbols until the length of the third frame matches the length of the second frame. In this case, the increase in FEC coding efficiency is sufficient to compensate for the latency incurred by padding.

15 The embodiments of the invention, including an embodiment with concatenated RS and turbo codes, use interleaving during the encoding process and deinterleaving during the decoding process. In the embodiment of FIG. 15, the interleaving operation occurs during the packing of the second blocks from the second frame into the third blocks of 20 the third frame at step 1508, and vice-versa. Again, however, the interleaving process can occur as a separate step from the packing process and still fall within the scope of the invention. Although byte interleaving is preferable, the interleaving operation can be either bit interleaving or byte interleaving.

25 FIG. 16 is a block flow diagram of a decoding process 600 in accordance with an embodiment of the invention including concatenated RS and turbo codes. The turbo-code error correcting code and the third blocks are recovered from the fourth frame at step 1602. The turbo-code

error correcting code is used to correct errors for the third blocks at step 1604. The second blocks are unpacked from the third blocks at step 1606. The unpacking process also includes a deinterleaving operation. The RS error correcting codes and the first blocks are recovered from the second 5 blocks at step 1608. The RS error correcting codes are used to correct errors for the first blocks at step 1610.

FIG. 17 illustrates an exemplary embodiment of how code blocks are packed into a fourth frame in the encoding step for an embodiment including concatenated RS and TPC codes. For ease of explanation, the 10 illustrated embodiment includes concatenation of a RS code with a two-dimensional TPC code. It is to be understood, however, that a variety of RS and turbo-codes may be concatenated in a manner consistent with the invention.

The illustrated exemplary embodiment also includes full byte-interleaving in packing blocks from the second frame into the third frame. The degree of interleaving effects system performance because higher levels of interleaving result in greater decorrelation of errors between the RS and TPC codes. The level of interleaving, however, also affects latency, processing time, and overall system complexity and cost. Thus, the 20 benefits of a selected interleaving depth should be balanced against the associated cost in terms of system complexity. As discussed above, byte interleaving is generally sufficient to achieve an appropriate performance/complexity balance. It is to be understood, however, that the present invention is not limited to any particular interleaving approach. In 25 fact, embodiments of the present invention may include full or partial byte interleaving or full or partial bit interleaving.

In the exemplary embodiment illustrated in FIG. 17, an integral number  $k_1$  of first blocks 1702 at the first (outer) encoding level are packed

- into a first frame 1704 (i.e., the unencoded outer frame). Check symbols 1706 for first blocks 1702 are generated by an RS encoder (e.g., first encoder 204) of a FEC encoder (e.g., FEC codec 104 or FEC encoder 200). Check symbols 1706 are appended to first blocks 1702 to form second blocks 1708.
- 5 Second blocks 1708 are packed into a second frame 1710 (i.e., the encoded outer frame). In the illustrated exemplary embodiment, each of the k2 bytes from each of the second blocks 1708 are interleaved, and they are packed into third blocks 1714 of a third frame 1712 (i.e., unencoded inner frame). In one embodiment, interleaving of the bytes may be performed,
- 10 for example, as illustrated in FIG. 8.

In this example, the second frame 1710 and third frame 1712 have the same length in terms of bits (or bytes), although the block sizes will likely vary between the two frames. In other words, the third frame 1712 is required to be an integral number of third blocks 1714, the size of which is 15 different from that of second blocks 1708. Thus, in order for second frame 1710 and third frame 1712 to be of the same length, the number of second blocks 1708 and third blocks 1714 per frame in each of these frames, respectively, has to be chosen appropriately.

If second frame 1710 and third frame 1712 cannot be made to match 20 with an integral number of blocks, third frame 1712 may be padded or “stuffed” with dummy symbols until they are of equal length. The padding process, however, represents an increase in latency in a hardware implementation, or increased processing time in software. In one embodiment of the invention, the lengths of the frames are therefore 25 chosen to minimize the number (or reduce to zero) of stuffed symbols, while at the same time keeping the number of second blocks per second frame to a minimum.

Once second blocks 1708 from second frame 1710 are packed and interleaved into third blocks 1714 of third frame 1712, the third frame is encoded with the TPC code by the TPC encoder (e.g., second encoder 208) of an FEC encoder (e.g., FEC codec 104 or FEC encoder 200). Those skilled in the art will recognize that an exemplary two-dimensional TPC encoder may include a combination of two simple encoders, e.g., recursive convolutional encoders, with an interleaver therebetween. A block of information bits provided at the input of the TPC encoder are transmitted unencoded, along with check symbols generated by the two simple encoders. A first group of check symbols are generated by the first encoder based on the information bits. The information bits are then permuted by the interleaver before being provided to the second encoder. The second encoder produces check symbols based on the interleaved information bits. A variety of commercially available TPC encoders will be known to those skilled in the art.

In the illustrated exemplary embodiment, the TPC encoder produces a fourth frame 1720 using the blocks 1714 of the third frame as the encoder input. The fourth frame may be illustrated as a block having an information bit portion 1722, with three separate check bit portions, 1724, 1726, 1728. In the case of full byte interleaving, as shown in FIG. 17, the information bit portion may include the bits of the third frame distributed in  $k_1$  columns and  $k_2$  rows. The first check bit portion 1724 includes a check bits row associated with each of the  $k_2$  information bit rows, and the second check bit portion 1726 includes a check bit column associated with each of the  $k_1$  information bit columns. The third check bit portion 1728 includes check bits derived from the column 1726 and/or row 1724 check bits. Once the two-level encoding process is performed, the encoded data

stream is communicated to a transceiver (e.g., transceiver 108) for decoding by a FEC decoder (e.g., FEC codec 110 or FEC decoder 300).

Turning now to FIG. 18, there is provided a plot of the results of the software-based Monte-Carlo simulation described above comparing coding gains for various concatenated RS embodiments with concatenated RS and TPC embodiments. In the legend for FIG. 18, the TPC codes are identified in the format “TPC(x,y)” wherein “x” is the total number of columns in the fourth frame, including information bit and check bit columns, and “y” is the total number of information bit columns. The RS codes are identified by (n/k) designation as described above.

The curve 1802 illustrates the coding gain associated with use of a TPC(62,57) code alone, and curve 1804 illustrates coding gain associated with use of concatenated RS(223,207) and RS (255,223) codes. As shown, use of a TPC alone code, as opposed to concatenated RS codes, has the result of shifting the “knee” of the coding gain curve to the left in the illustrated plot, i.e. the relationship of  $\text{BER}_{(\text{out})}$  vs.  $Q_{\text{in}}(\text{dB})$  is pushed to lower values of  $Q_{\text{in}}(\text{db})$ . This significant gain at high BER is, however, slightly offset by reduced performance at lower BER, as indicated by the apparent “flaring” or reduced slope of the curve 1802.

Advantageously, however, concatenation of a RS code and a TPC code produces a coding gain curve that falls off faster, i.e., has increased slope, compared to use of a TPC code alone. This reflects improved performance at lower BER. For example, curve 1806 illustrates performance of concatenated RS(255,247) and TPC(64,57) codes. As shown, concatenation of the RS and TPC codes produces a curve 1806 with increased slope compared to curve 1802, which represents performance of a TPC code alone.

It will be appreciated that the functionality described for the embodiments of the invention may be implemented in hardware, software, or a combination of hardware and software, using well-known signal process techniques. If in software, a processor and machine-readable medium is required. The processor can be any type of processor capable of providing the speed and functionality required by the embodiments of the invention. For example, the processor could be a process from the Pentium® family of processors made by Intel Corporation, or the family of processors made by Motorola. Machine-readable media include any media capable of storing instructions adapted to be executed by a processor. Some examples of such media include, but are not limited to, read-only memory (ROM), random-access memory (RAM), programmable ROM (PROM), erasable programmable ROM (EPROM), electronically erasable programmable ROM (EEPROM), dynamic RAM (DRAM), magnetic disk (e.g., floppy disk and hard drive), optical disk (e.g., CD-ROM), and any other device that can store digital information.

In one embodiment, the instructions are stored on the medium in a compressed and/or encrypted format. As used herein, the phrase "adapted to be executed by a processor" is meant to encompass instructions stored in a compressed and/or encrypted format, as well as instructions that have to be compiled or installed by an installer before being executed by the processor. Further the processor and machine-readable medium may be part of a larger system that may contain various combinations of machine-readable storage devices through various I/O controllers, which are accessible by the processor and which are capable of storing a combination of computer program instructions and data.

Finally, in another example, the embodiments were described in a communication network. A communication network, however, can utilize

an infinite number of network devices configured in an infinite number of ways. The communication network described herein is merely used by way of example, and is not meant to limit the scope of the invention.